Generative Adversarial Networks

Deep Learning (DSE316/616)

Vinod K Kurmi

Assistant Professor, DSE

Indian Institute of Science Education and Research Bhopal

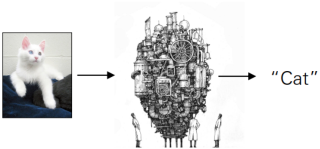
Oct 27, 2022

Disclaimer

- Much of the material and slides for this lecture were borrowed from
    - Bernhard Schölkopf's MLSS 2017 lecture,
    - Tommi Jaakkola's 6.867 class,
    - CMP784: Deep Learning Fall 2021 Erkut Erdem Hacettepe University
    - Fei-Fei Li, Andrej Karpathy and Justin Johnson's CS231n class
    - Hongsheng Li's ELEG5491 class
    - Tsz-Chiu Au slides
    - Mitesh Khapra Class notes

Discriminative vs. Generative Models

$$p(y|x)$$

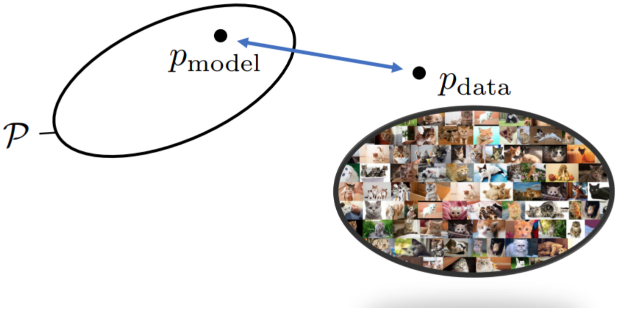


Discriminative models

$$p(x|y)$$

Generative models

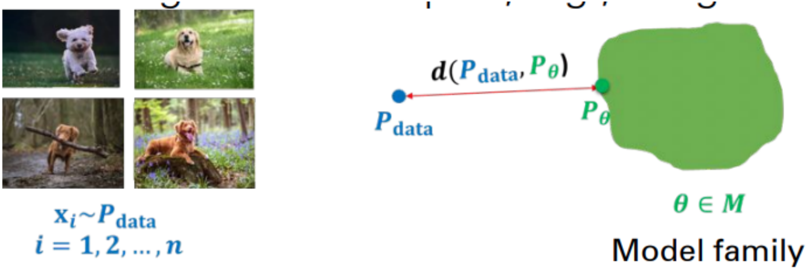
Generative Modeling



- **Goal:** Learn some underlying hidden structure of the training samples to generate novel samples from same data distribution

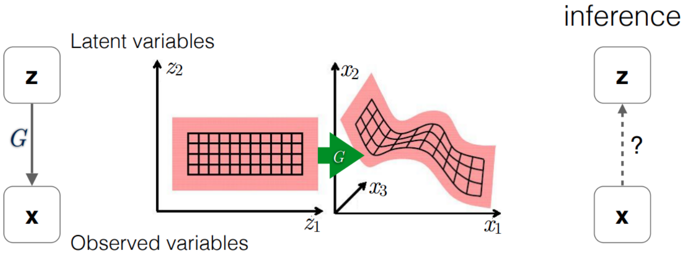
Learning a generative model

- We are given a training set of examples, e.g., images of dogs

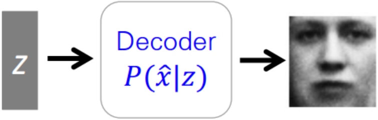


- We want to learn a probability distribution $p(x)$ over images x s.t.
  - **Generation:** If we sample x_{new} $p(x)$, x_{new} should look like a dog (sampling)
  - **Density estimation:** $p(x)$ should be high if x looks like a dog, and low otherwise (anomaly detection)
  - **Unsupervised representation learning:** We should be able to learn what these images have in common, e.g., ears, tail, etc. (features)

Latent variable model

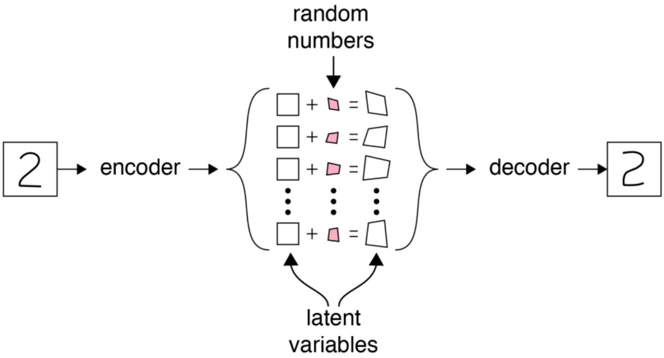


Sample from g(z)
e.g. Standard
Gaussian

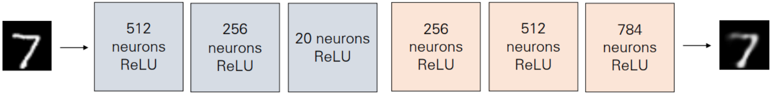


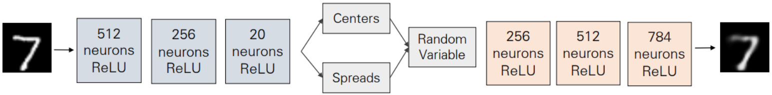
$\boxed{z \sim g(z)}$ $\boxed{\hat{x} = f(z)}$ $\boxed{\hat{x} \sim P(x|z)}$

Variational Autoencoders



Variational Autoencoders





Variational AE

$$
\begin{aligned}
\log p_\theta(x^{(i)}) &= \mathbf{E}_{z\sim q_\phi(z|x^{(i)})}\left[\log p_\theta(x^{(i)})\right] \quad (p_\theta(x^{(i)}) \text{ Does not depend on } z) \\
&= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z)p_\theta(z)}{p_\theta(z \mid x^{(i)})}\right] \quad (\text{Bayes' Rule}) \\
&= \mathbf{E}_z\left[\log \frac{p_\theta(x^{(i)} \mid z)p_\theta(z)}{p_\theta(z \mid x^{(i)})} \frac{q_\phi(z \mid x^{(i)})}{q_\phi(z \mid x^{(i)})}\right] \quad (\text{Multiply by constant}) \\
&= \mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - \mathbf{E}_z\left[\log \frac{q_\phi(z \mid x^{(i)})}{p_\theta(z)}\right] + \mathbf{E}_z\left[\log \frac{q_\phi(z \mid x^{(i)})}{p_\theta(z \mid x^{(i)})}\right] \quad (\text{Logarithms}) \\
&= \underbrace{\mathbf{E}_z\left[\log p_\theta(x^{(i)} \mid z)\right] - D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z))}_{\mathcal{L}(x^{(i)},\theta,\phi)\ \text{"Elbow"}} + \underbrace{D_{KL}(q_\phi(z \mid x^{(i)}) \,||\, p_\theta(z \mid x^{(i)}))}_{\geq 0}
\end{aligned}
$$

$$\log p_\theta(x^{(i)}) \geq \mathcal{L}(x^{(i)}, \theta, \phi)$$

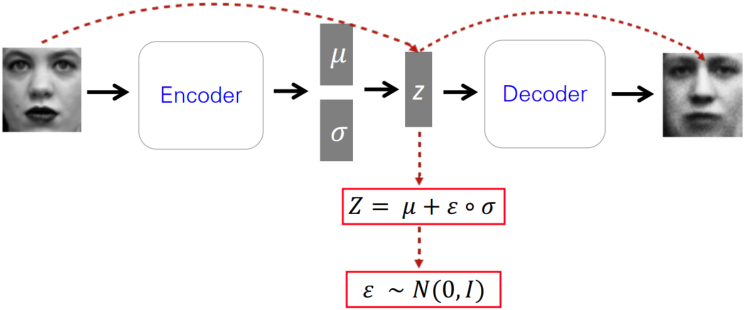
Variational lower bound (elbow)

$$\theta^*, \phi^* = \arg\max_{\theta,\phi} \sum_{i=1}^{N} \mathcal{L}(x^{(i)}, \theta, \phi)$$

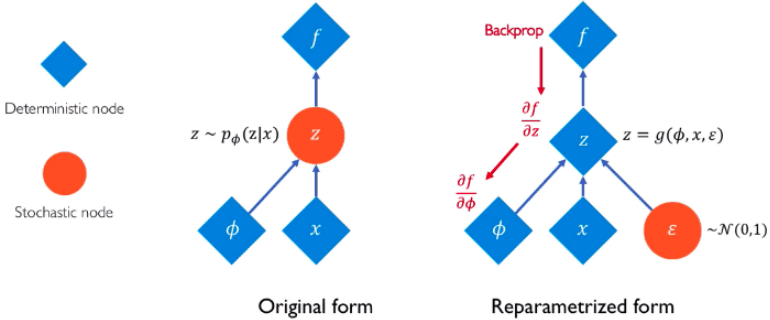
Training: Maximize lower bound

55

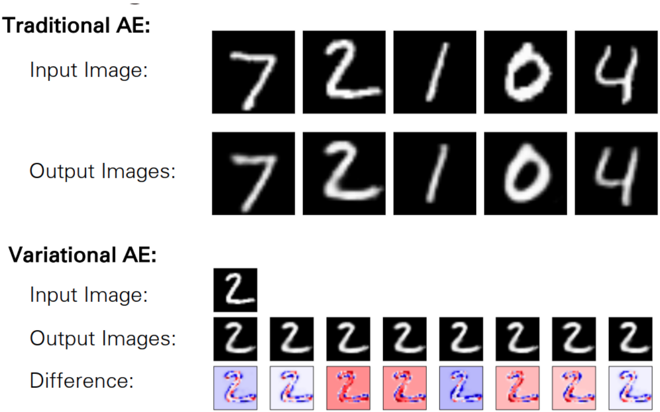
Reparametrization Trick



Reparametrization Trick



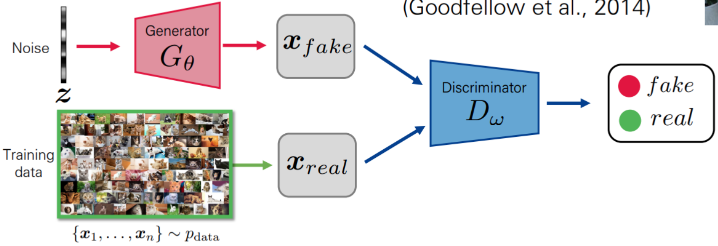
Training VAE



Genetive Adversarial Networks (GANs)

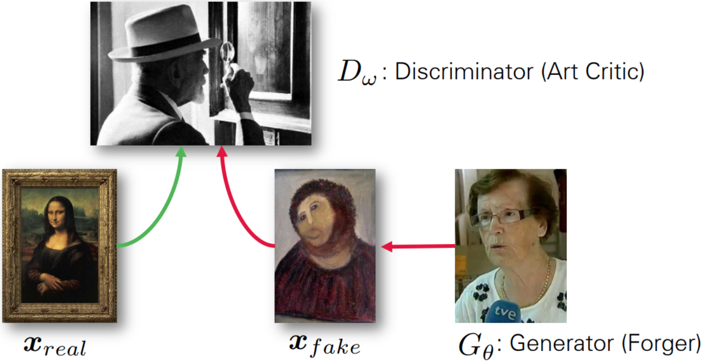
Genetive Adversarial Networks (GANs)

(Goodfellow et al., 2014)



- A game between a generator $G_\theta(z)$ and a discriminator $D_\omega(x)$
  - Generator tries to fool discriminator (i.e. generate realistic samples)
  - Discriminator tries to distinguish fake from real samples

Intuition behind GANs



GAN Training: Minimax Game

$$\min_{\theta} \max_{\omega} \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}} [\log D_{\omega}(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}} [\log (1 - D_{\omega}(G_{\theta}(\boldsymbol{z})))]$$

Real data

Noise vector used to generate data

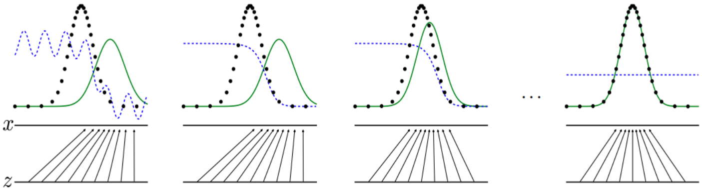
$$J^{(D)} = -\frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}} \log D(\boldsymbol{x}) - \frac{1}{2}\mathbb{E}_{\boldsymbol{z}} \log (1 - D(G(\boldsymbol{z})))$$

Cross-entropy loss for binary classification

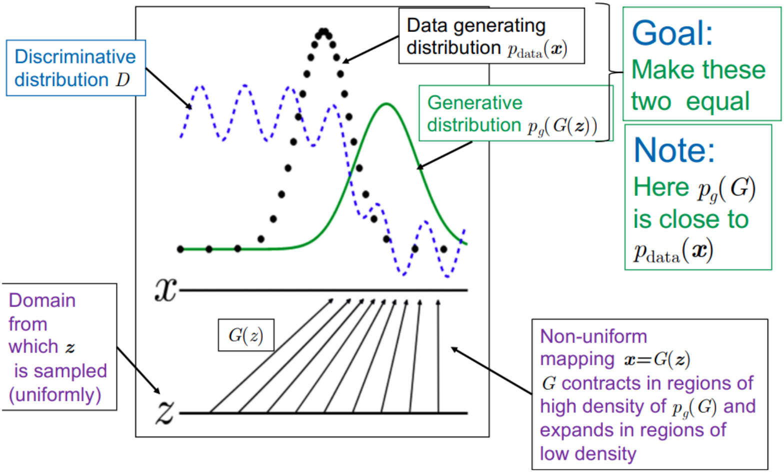
$$J^{(G)} = -\frac{1}{2}\mathbb{E}_{\boldsymbol{z}} \log D(G(\boldsymbol{z}))$$

Generator maximizes the log-probability of the discriminator being mistaken

Training Procedure

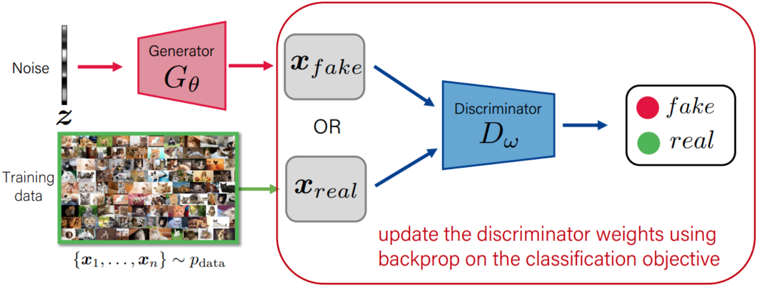


Training Procedure



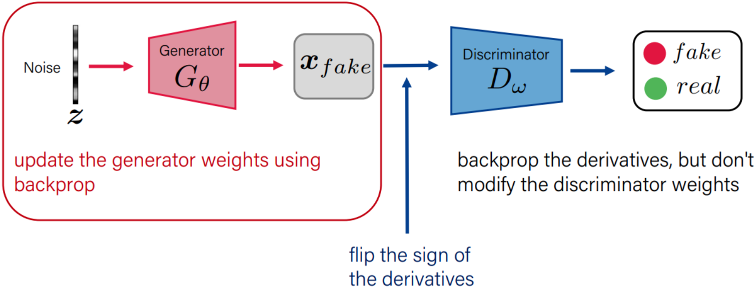
Training Procedure

- Updating the discriminator:



Training Procedure

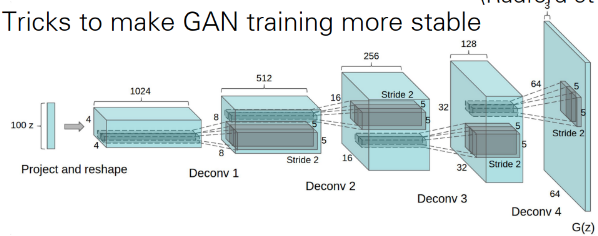
- Updating the generator:



Deep Convolutional GANs (DCGAN)

(Radford et al., 2015)

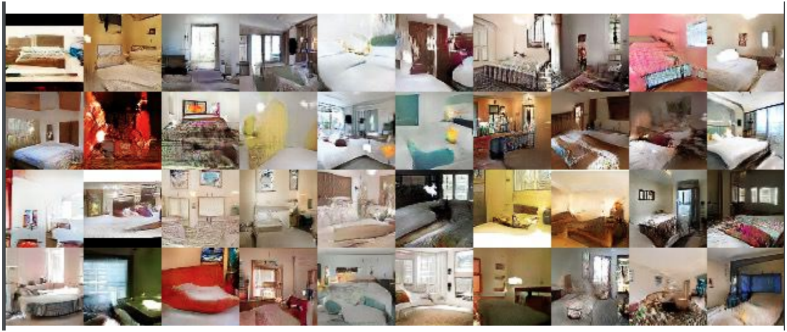
- **Idea:** Tricks to make GAN training more stable



- No fully connected layers
- Batch Normalization (Ioffe and Szegedy, 2015)
- Leaky Rectifier in D
- Use Adam (Kingma and Ba, 2015)
- Tweak Adam hyperparameters a bit (lr=0.0002, b1=0.5)

24

Deep Convolutional GANs (DCGAN)

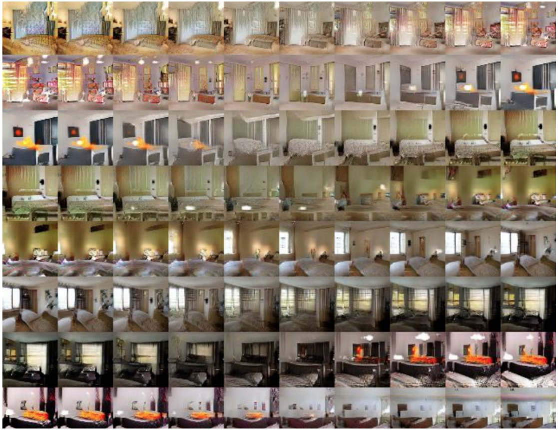


Deep Convolutional GANs (DCGAN)

Walking over the latent space

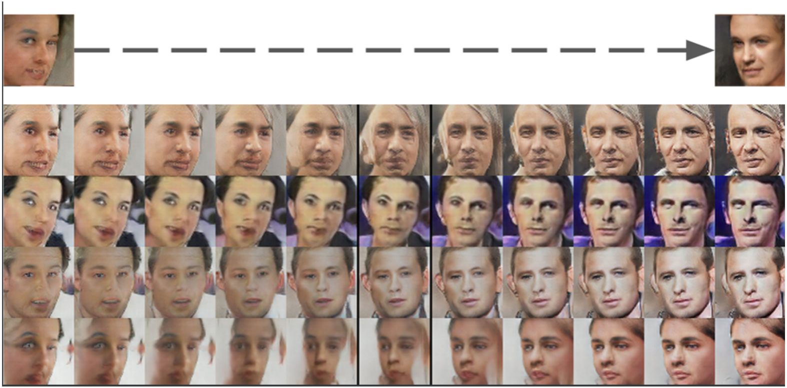
(Radford et al., 2015)

- Interpolation suggests non-overfitting behavior



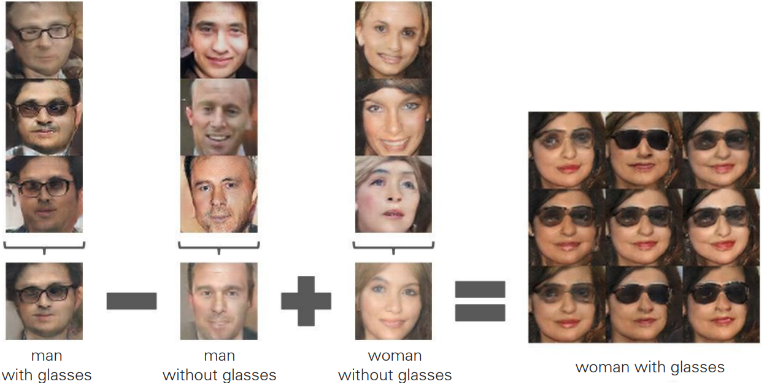
Deep Convolutional GANs (DCGAN)

- Walking over the latent space



Deep Convolutional GANs (DCGAN)

- Vector Space Arithmetic



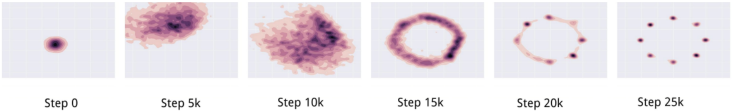
GAN Failures: Mode Collapse

$$\min_G \max_D V(G, D) \neq \max_D \min_G V(G, D)$$

- **D** in inner loop: convergence to correct distribution
- **G** in inner loop: place all mass on most likely point

Mode Collapse: Solutions

- **Unrolled GANs** (Metz et al 2016): Prevents mode collapse by backproping through a set of (k) updates of the discriminator to update generator parameters

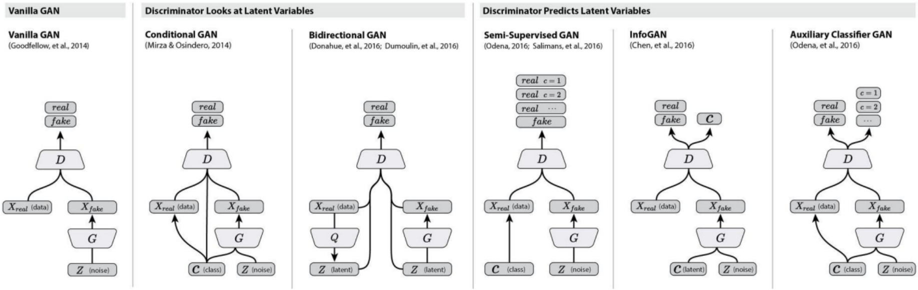


- **VEEGAN** (Srivastava et al 2017): Introduce a reconstructor network which is learned both to map the true data distribution p(x) to a Gaussian and to approximately invert the generator network.

GAN Evaluation

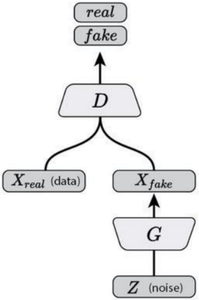
- Quantitatively evaluating GANs is not straight forward:
  - Max Likelihood is a poor indication of sample quality
- Some evaluation metrics
  - Inception Score (IS):
    - y = labels given gen. image. p(y|x) is from classifier - InceptionNet
    - $IS(p_g) = e^{E_{x \in P_g} KL(P_m(y|x)||p(y))}$
  - Fréchet inception distance (FID): (Currently most popular)
    - Estimate mean m and covariance C from classifier output - InceptionNet
    - $$d^2((\boldsymbol{m}, \boldsymbol{C}), (\boldsymbol{m}_w, \boldsymbol{C}_w)) = \|\boldsymbol{m} - \boldsymbol{m}_w\|_2^2 + \mathrm{Tr}\left(\boldsymbol{C} + \boldsymbol{C}_w - 2\left(\boldsymbol{C}\boldsymbol{C}_w\right)^{1/2}\right)$$
  - Kernel MMD (Maximum Mean Discrepancy):
    - $$\mathrm{MMD}(\mathbb{P}_r, \mathbb{P}_g) = \left(\mathbb{E}_{\substack{\mathbf{x}_r, \mathbf{x}_r' \sim \mathbb{P}_r, \\ \mathbf{x}_g, \mathbf{x}_g' \sim \mathbb{P}_g}}\left[k(\mathbf{x}_r, \mathbf{x}_r') - 2k(\mathbf{x}_r, \mathbf{x}_g) + k(\mathbf{x}_g, \mathbf{x}_g')\right]\right)^{\frac{1}{2}}$$

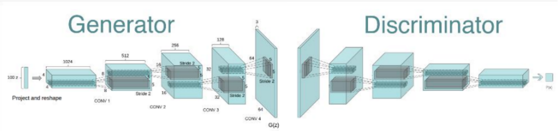
Subclasses of GANs



Vanilla GAN (Goodfellow et al., 2014

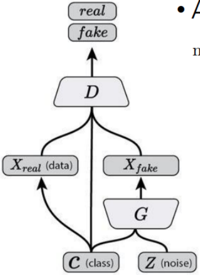
- Updating the generator:





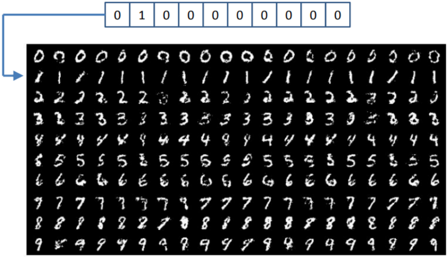
DCGAN (Radford et al., 2015)

Conditional GAN (Mirza et al)



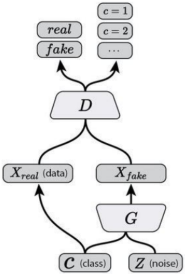
- Add conditional variables $\boldsymbol{y}$ into G and D

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x}|\boldsymbol{y})] + \mathbb{E}_{\boldsymbol{z}\sim p_z(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z}|\boldsymbol{y})))]$$



3

Auxiliary Classifier GAN(Odena et al)



- Every generated sample has a corresponding class label

$$L_S = E[\log P(S = real \mid X_{real})] + E[\log P(S = fake \mid X_{fake})]$$
$$L_C = E[\log P(C = c \mid X_{real})] + E[\log P(C = c \mid X_{fake})]$$

- D is trained to maximize $L_S + L_C$
- G is trained to maximize $L_C - L_S$

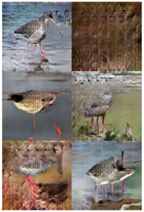
- Learns a representation for z that is independent of class label

Auxiliary Classifier GAN

monarch butterfly

goldfinch

daisy



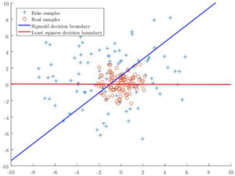
redshank

grey whale

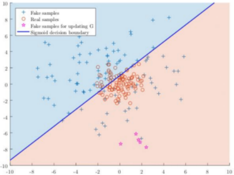
Least Squares GAN (LSGAN)

- Use a loss function that provides smooth and non-saturating gradient in discriminator D

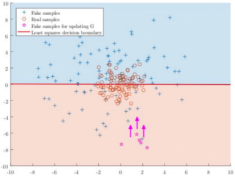
$$\min_D V_{\text{LSGAN}}(D) = \frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}\left[(D(\boldsymbol{x}) - b)^2\right] + \frac{1}{2}\mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}\left[(D(G(\boldsymbol{z})) - a)^2\right]$$
$$\min_G V_{\text{LSGAN}}(G) = \frac{1}{2}\mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}\left[(D(G(\boldsymbol{z})) - c)^2\right],$$



Decision boundaries of Sigmoid & Least Squares loss functions



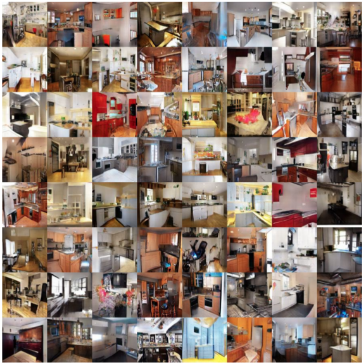
Sigmoid decision boundary



Least Squares decision boundary

Least Squares GAN (LSGAN)

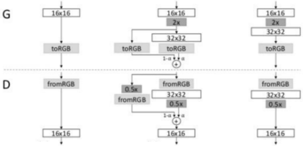
Church

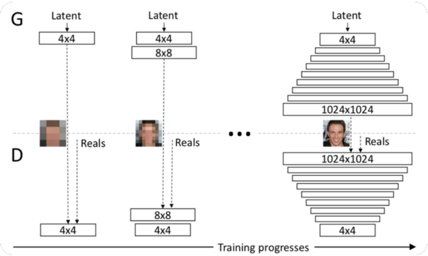


Kitchen

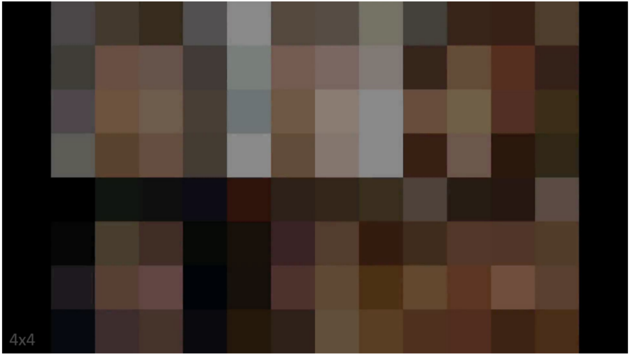
Progressive GANs(Karras et al., 2018)

- Progressively generate high-res images
- Multi-step training from low to high resolutions





Progressive GANs(Karras et al., 2018)



- Training process

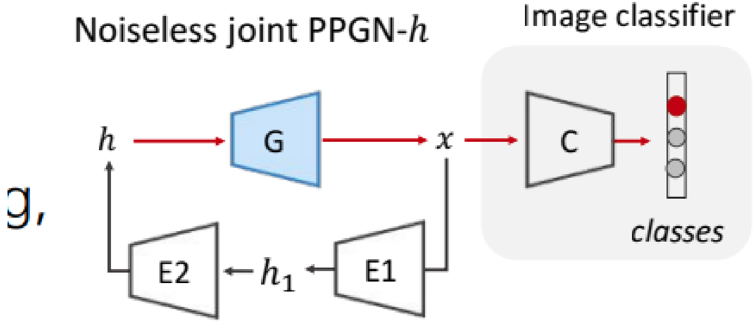
Progressive GANs(Karras et al., 2018)

GANs

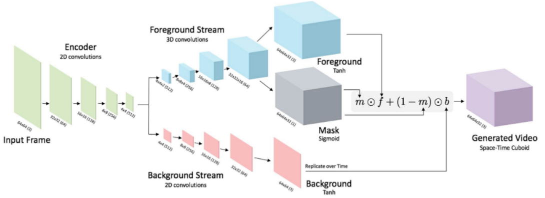
Next Class

Class-specific Image Generation (Nguyen et al., 2016)

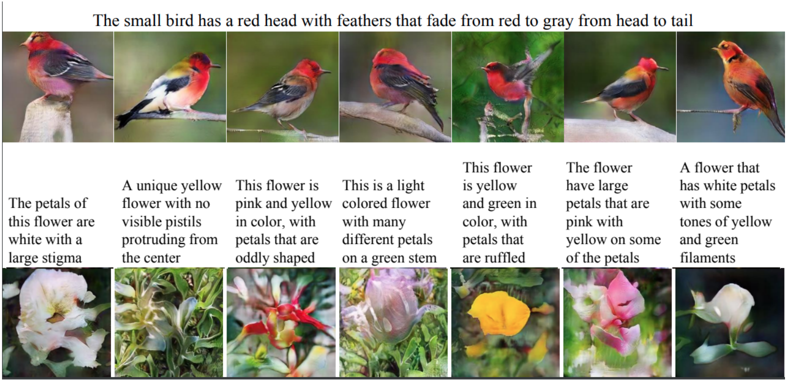
- Generates 227x227 realistic images from all ImageNet classes
- Combines adversarial training, moment matching, denoising autoencoders, and Langevin sampling



Video Generation (Vondrick et al., 2016)



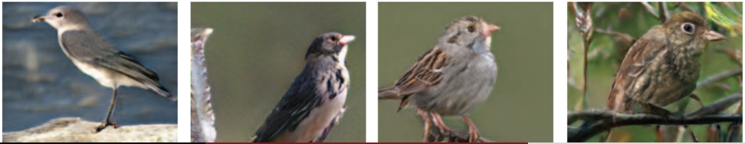
Text-to-Image Synthesis (Zhang et al., 2016)



Text-to-Image Synthesis (Zhu et al., 2019)

This bird has a white throat and a dark yellow bill and grey wings.	This particular bird has a belly that is yellow and brown.	This bird is a lime green with greyish wings and long legs.	This yellow bird has a thin beak and jet black eyes and thin feet.

This bird has wings that are grey and has a white belly.	This bird has wings that are black and has a white belly.	This is a grey bird with a brown wing and a small orange beak.	This bird has a short brown bill, a white eyering, and a medium brown crown.



Single Image Super-Resolution ((Ledig et al., 2016))

- Combine content loss with adversarial loss

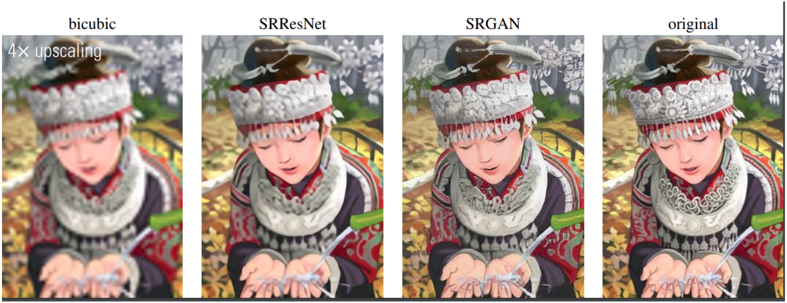


Image Inpainting (Pathak et al., 2016)

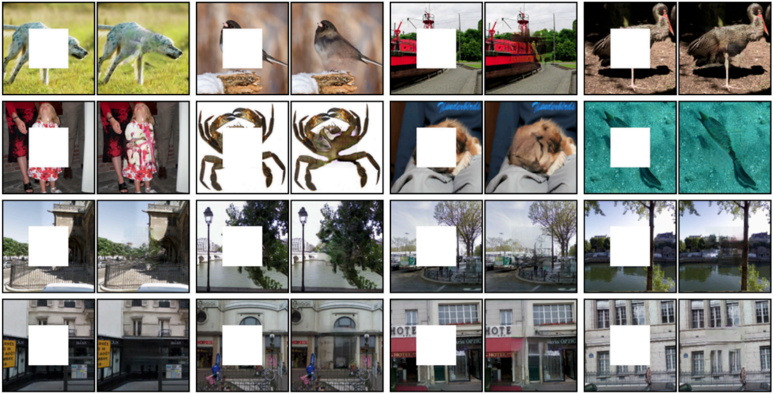


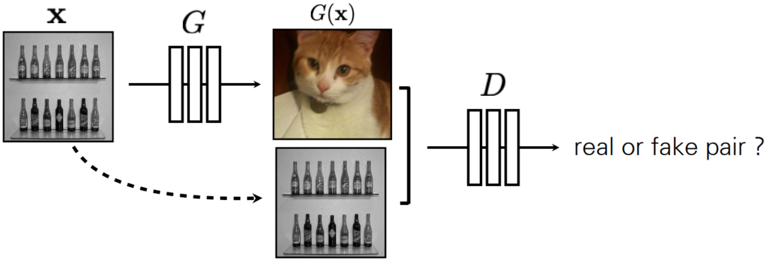
Image to Image Translation (Pix2Pix)

(Isola et al. 2016)

Image to Image Translation (Pix2Pix)

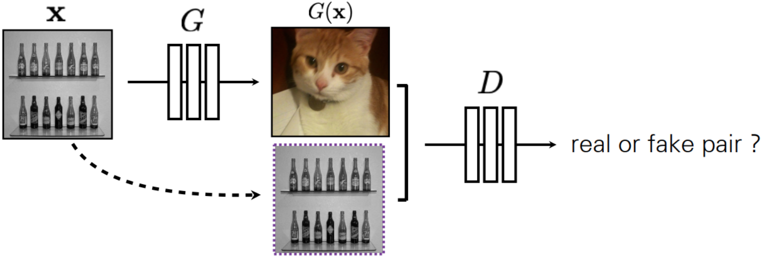
(Isola et al. 2016)

Image to Image Translation (Pix2Pix)



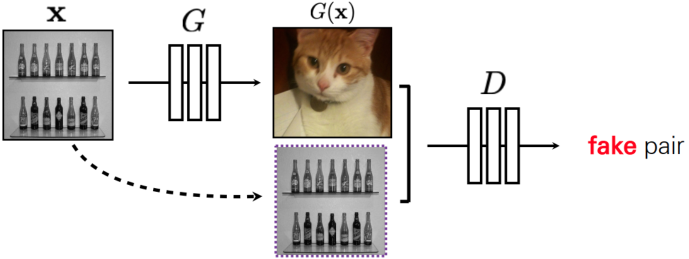
$$\arg\min_{G}\max_{D}\ \mathbb{E}_{\mathbf{x},\mathbf{y}}[\ \log D(G(\mathbf{x})) \ + \ \log(1 - D(\mathbf{y}))\ ]$$

Image to Image Translation (Pix2Pix)



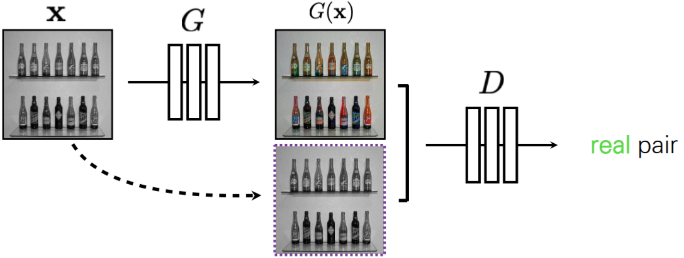
$$\arg\min_{G}\max_{D}\ \mathbb{E}_{\mathbf{x},\mathbf{y}}[\ \log D(\boxed{\mathbf{x}}, G(\mathbf{x})) + \log(1 - D(\boxed{\mathbf{x}}, \mathbf{y}))\ ]$$

Image to Image Translation (Pix2Pix)



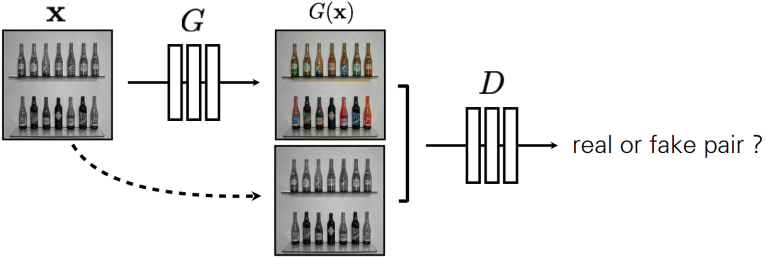
$$\arg\min_G \max_D \ \mathbb{E}_{\mathbf{x},\mathbf{y}}[\ \log D(\boxed{\mathbf{x}}, G(\mathbf{x})) + \log(1 - D(\boxed{\mathbf{x}}, \mathbf{y}))\ ]$$

Image to Image Translation (Pix2Pix)



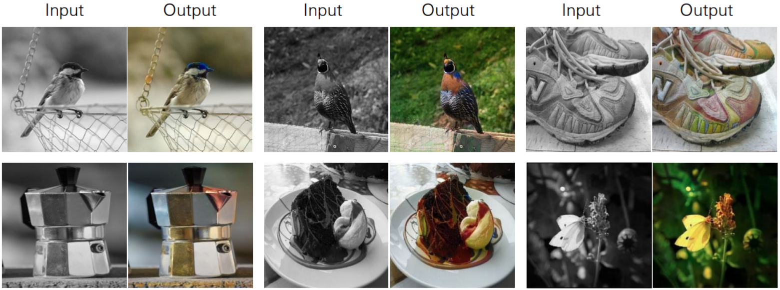
$$\arg\min_{G}\max_{D}\ \mathbb{E}_{\mathbf{x},\mathbf{y}}\left[\ \log D(\boxed{\mathbf{x}}, G(\mathbf{x})) + \log(1 - D(\boxed{\mathbf{x}}, \mathbf{y}))\ \right]$$

Image to Image Translation (Pix2Pix)

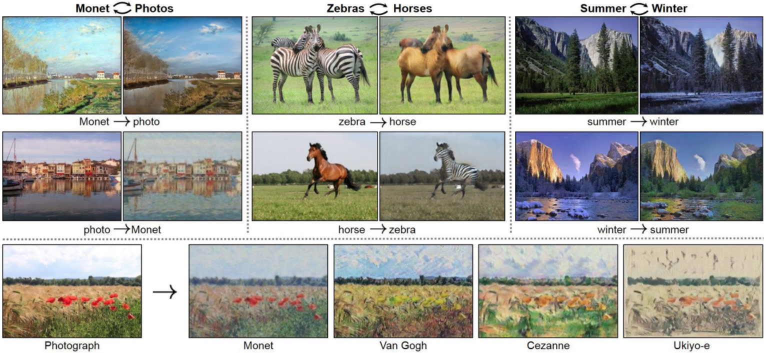


$$\boxed{\arg\min_G \max_D \ \mathbb{E}_{\mathbf{x},\mathbf{y}}[\ \log D(\mathbf{x}, G(\mathbf{x})) + \log(1 - D(\mathbf{x},\mathbf{y}))\ ]}$$

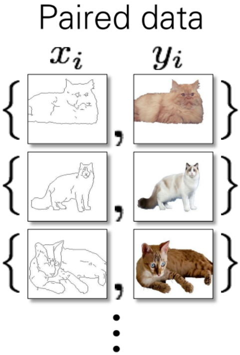
BW –> Color

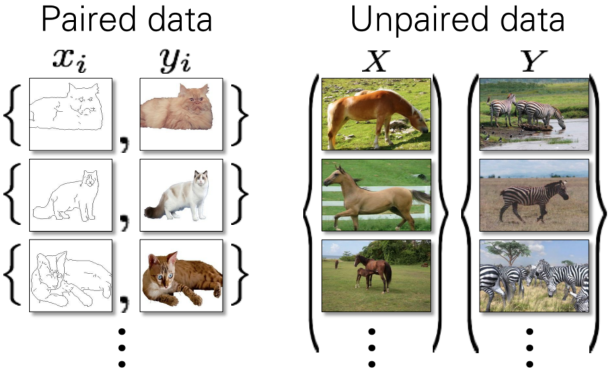


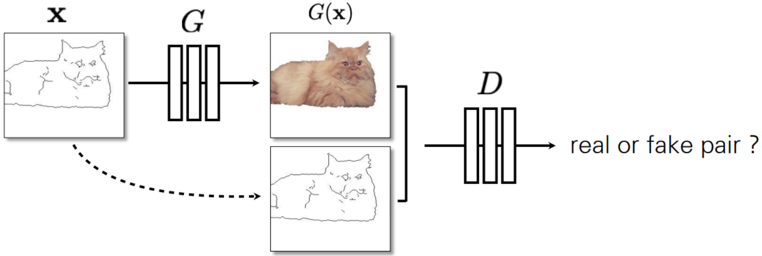
CycleGAN: Pix2Pix w/o input-output pairs



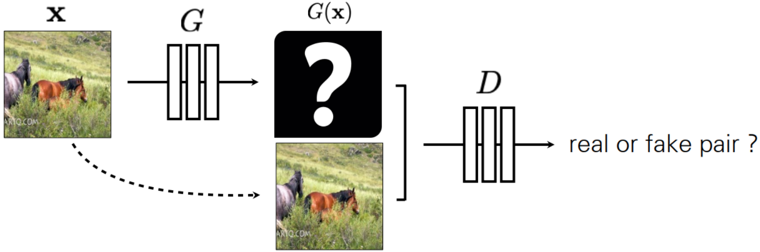
(Zhu et al. 2017)


Paired data
x_i
y_i


Paired data
x_i
y_i
Unpaired data
X
Y

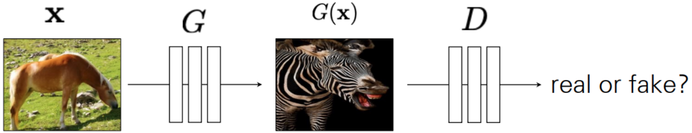


$$\arg\min_G \max_D \ \mathbb{E}_{\mathbf{x},\mathbf{y}}[\ \log D(\mathbf{x}, G(\mathbf{x})) + \log(1 - D(\mathbf{x}, \mathbf{y}))\ ]$$



$$\arg\min_G \max_D \ \mathbb{E}_{\mathbf{x},\mathbf{y}}[\ \log D(\mathbf{x}, G(\mathbf{x})) + \log(1 - D(\mathbf{x}, \mathbf{y}))\ ]$$

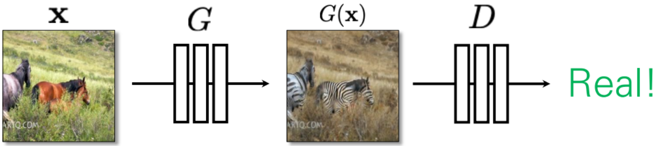
No input-output pairs!

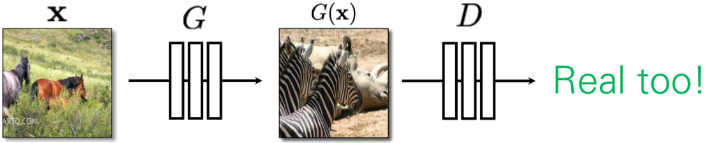


$$\arg\min_{G}\max_{D}\ \mathbb{E}_{\mathbf{x},\mathbf{y}}[\ \log D(G(\mathbf{x}))\ \ +\ \ \log(1 - D(\mathbf{y}))\ ]$$

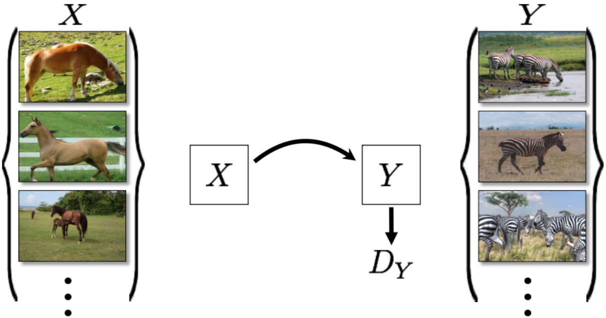
Usually loss functions check if output matches a target instance

GAN loss checks if output is part of an admissible set

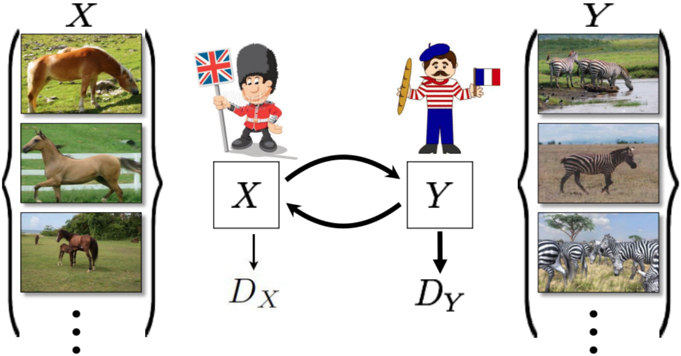

$\mathbf{x}$
G
$G(\mathbf{x})$
D
Real!



Nothing to force output to correspond to input

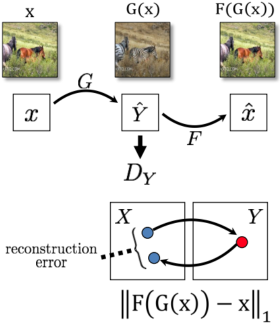


[Zhu et al. 2017], [Yi et al. 2017], [Kim et al. 2017]

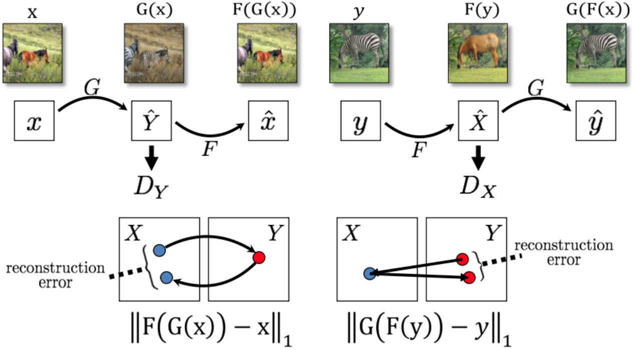


X
Y
X
Y
D_X
D_Y

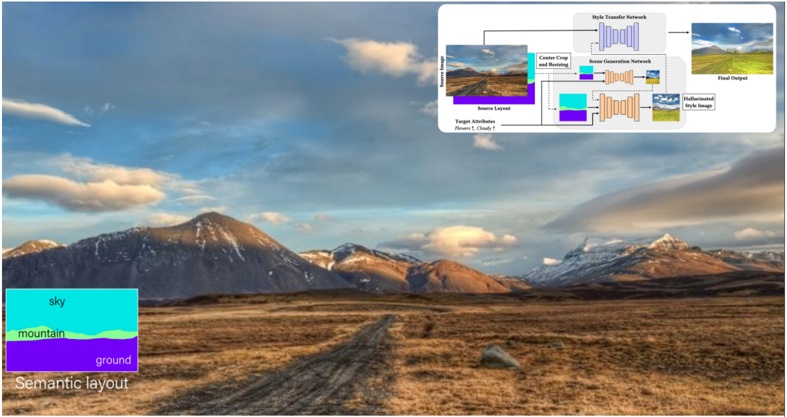
Cycle-Consistent Adversarial Networks



Cycle-Consistent Adversarial Networks



Cycle Consistency Loss

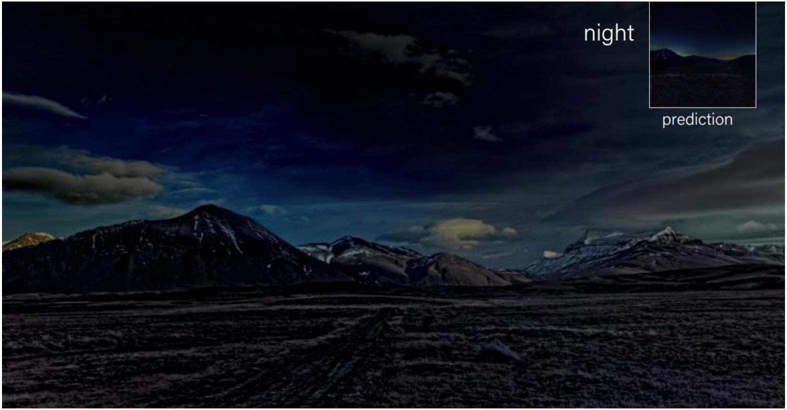


Manipulating Attributes of Natural Scenes via Hallucination [Karacan et al., 2020]

Cycle Consistency Loss

Manipulating Attributes of Natural Scenes via Hallucination [Karacan et al., 2020]

Semantic Image Synthesis (SPADE)

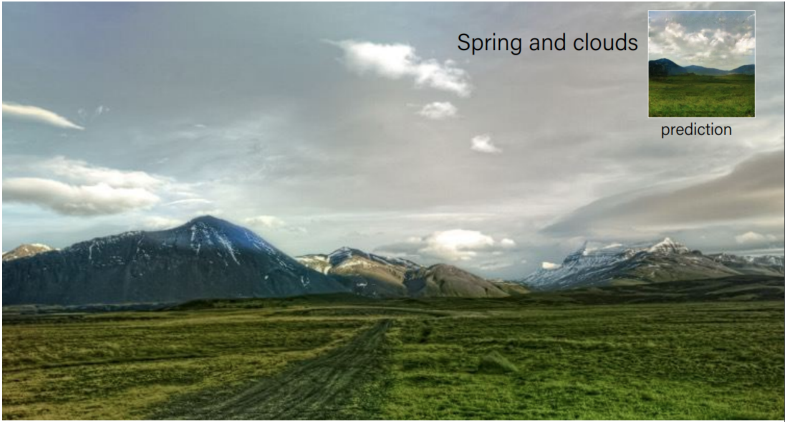


Manipulating Attributes of Natural Scenes via Hallucination [Karacan et al., 2020]

Manipulating Attributes of Natural Scenes via Hallucination [Karacan et al., 2020]

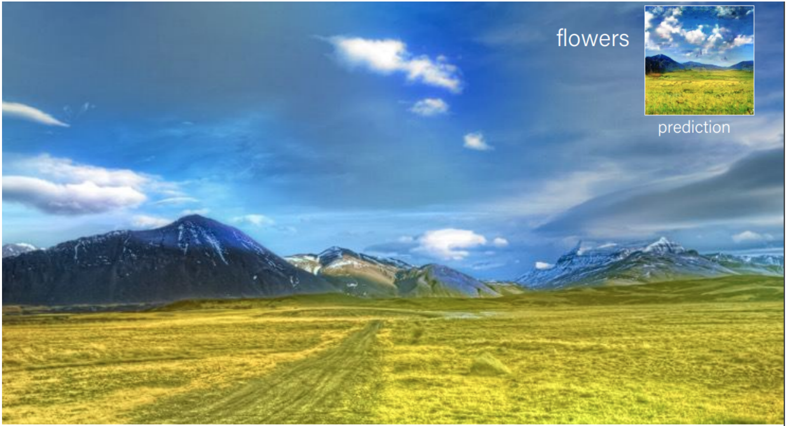
Manipulating Attributes of Natural Scenes via Hallucination [Karacan et al., 2020]

Manipulating Attributes of Natural Scenes via Hallucination [Karacan et al., 2020]



Manipulating Attributes of Natural Scenes via Hallucination [Karacan et al., 2020]

Manipulating Attributes of Natural Scenes via Hallucination [Karacan et al., 2020]



Manipulating Attributes of Natural Scenes via Hallucination [Karacan et al., 2018]